ROUTING IN CLUSTERED MULTIHOP, MOBILE WIRELESS NETWORKS WITH FADING CHANNEL

Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, Mario Gerla
Computer Science Department
University of California, Los Angeles
CA 90024, USA

A clusterhead-token infrastructure for multihop, mobile wireless networks has been designed. Traditional routing algorithms in wireline networks are not feasible for mobile wireless environment due to the dynamic change in link connectivity. To gain better performance for clustered multihop, mobile wireless networks, routing must take into account radio channel access, code scheduling, and channel reservation. In this paper, we propose some heuristic routing schemes for clustered multihop, mobile wireless networks. A packet delay improvement up to fourfold has been observed in our simulations compared with shortest-path scheme, making multimedia traffic viable. A radio channel model has been included to investigate the impact of channel fading on our protocols. To reduce the run time, a parallel simulator has been designed. Speedups of up to tenfold have been observed on a 16 processor SP/2.

1 Introduction

Wireless networks provide mobile users with ubiquitous communicating capability and information access regardless of the location. In this paper we address a particular type of wireless networks called "multihop" networks. As a difference from "single hop" (i.e. cellular) networks [10] which require fixed base stations (sometimes called Mobile Support Stations, MSS, or Mobility Support Routers, MSR) interconnected by a wired backbone, multihop networks have no fixed based stations nor wired backbone [6]. The main motivation for mobile wireless multihopping is rapid deployment and dynamic reconfiguration. When the wireline network is not available, as in battlefield communications and search and rescue operations, multihop mobile wireless networks provide the only feasible means for ground communications and information accesses. Examples of such networks are ad-hoc networks [11,15] and packet radio networks [3,12]. The dynamic feature in multihop mobile wireless networks leads to the problem of keeping track of the topology connectivity [15]. Traditional routing protocols in single-hop mobile wireless networks [10] also have problems in multihop mobile wireless networks since there is no fixed home agent to maintain routing information. Due to the mobility of the hosts and the limit of wireless media, the problem of routing is complex. Frequent broadcasts of the routing table, or flooding, will degrade the throughput of channel access

and increase the overhead as the population of mobile hosts increases. Previous researches, including cluster TDMA [6] and cluster token (non-overlapped without clusterhead) [13], use different channel access and routing schemes for different infrastructures in the multihop mobile environment. In this paper, we propose a new infrastructure (clusterhead-token) and routing schemes which take advantage of cluster and channel access properties.

2 Simulation model

A multihop, mobile wireless network simulator is being built on an existing process-oriented, parallel simulation language called Maisie [1,14]. A Maisie program is a collection of entities, each of which represents a specific object in the physical system and may be created and destroyed dynamically. Entities communicate with each other using time-stamped messages. We generate 100 mobile hosts uniformly in a 1000x1000 pixel square area. One pixel represents 0.5 meter, and each mobile node ,which is represented by a Maisie entity, moves 1 meter/sec using a probability moving behavior. The radio transmission power is 100 pixels and the data rate is 2 Mb/s. One Maisie simulation clock represents $125\mu s$. We use this simulation model to evaluate the average delay of transmitting 100 packets (each packet size is 1KB) from one source node to a destination with an average 12 hops distance. The model configured with 100 nodes, 100 pixels transmission power,and 1 m/s mobility is called a sample case. Figure 2 shows the initial topology.

3 Radio channel model

An accurate radio channel simulation model is important not only for designing modulation and coding schemes that improve channel efficiency, but also for investigating the impact of channel fading on existing networking algorithms, such as clustering, routing and power adjustment. At present, most radio network protocol simulations are using the free space channel model which basically assumes that attenuation is only a function of transmitter-receiver distance. However, the radio channel characteristics are much too complex to be modeled by simple distance functions. Thus, the results are inaccurate.

To overcome this limitation, our simulator includes a rather sophisticated radio channel model which is an extension of the SIRCIM statistical impulse response model [16].

The radio channel is characterized by three propagation parameters: free space loss, multipath fading and shadowing. All these parameters are supplied by SIRCIM. SIRCIM provides impulse response characteristics which account

for multipath fading. SIRCIM outputs include for example: the distribution of the number of multipath components in a particular multipath delay profile; the distribution of the number of multipath components etc.

SIRCIM provides impulse response at the *signal* level which is suitable for radio designs. For network performance evaluation purposes, we are more interested in received power at the *packet* level. Assuming that the channel is a Direct Sequence-Spread Spectrum channel, we can derive the mean signal power by performing convolution of the spread spectrum random chip sequence with the impulse response of the simulated channel.

Furthermore, in a mobile radio environment, we must model the fluctuation of received power caused by change of positions. To this end, the correlation between received power at different positions must be known. SIRCIM provides only *small-scale* spatial and temporal correlations. We have augmented SIRCIM to account for *large-scale* correlation as well.

In addition to multipath fading, the SIRCIM accounts for the shadowing effect caused by diffraction of radio waves around sharp edges. Shadowing, the slow fading, has been characterized in the literatures by roughly a log-normal distribution, with a standard deviation that depends on the roughness. A common assumption is that shadowing is independent from one location to another. Unfortunately, this assumption is not valid in a dynamic model with mobile users,in which location dependent correlation must be accounted for. In our simulation, we include the correlation model for shadow attenuation developed in [9].

The SIRCIM channel module is invoked (with a function call) every time a packet is transmitted in the network. Repeated computation of attenuation is required because the nodes are mobile and therefore change their relative position continuously.

4 Clusterhead-token infrastructure

4.1 Clustering

In multihop, mobile wireless networks, the aggregation of nodes into clusters controlled by a cluster head provides a convenient framework for the development of important features such as code separation (among clusters), channel access, routing, and bandwidth allocation [6,8]. Using a distributed algorithm with a cluster, a node is elected to be the cluster head. All nodes within transmission range of the cluster head belong to this cluster. That is, all nodes in a cluster can communicate with a cluster head and (possibly) with each other. The complexity and overhead of clustering rests in the selection of the cluster

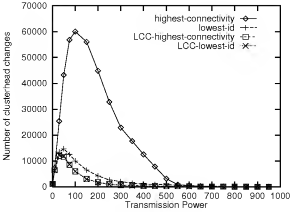

Figure 1: Cluster changes vs. Transmission Power

head. There are two possible distributed clustering algorithms, lowest-ID algorithm [4] and highest-connectivity (degree) algorithm [6]. The most important criterion is stability. Frequent cluster head changes adversely affect the performance of other protocols such as scheduling and allocation which rely on it. In our clustering algorithm (Least Cluster Change (LCC) clustering algorithm), only two conditions cause the cluster head to change. One is when two cluster heads come within range of each other, and the other is when a node becomes disconnected from any other cluster. This is an improvement (in stability) over existing algorithms which select the cluster head every time the cluster membership changes.

Following is our clustering algorithm specification.

1 : At the start we use lowest-id cluster algorithm or highest-connectivity cluster algorithm to create initial clusters.

2 : When a non-clusterhead node in cluster i move into a cluster j, no clusterhead in cluster i and j will be changed (only cluster members are changed).

3 : When a non-clusterhead node moves out its clusters and doesn't enter into any existing cluster, it becomes a new clusterhead, forming a new cluster.

4 : When clusterhead $C(i)$ from cluster i moves into the cluster j, it challenges the corresponding clusterhead $C(j)$. Either $C(i)$ or $C(j)$ will give

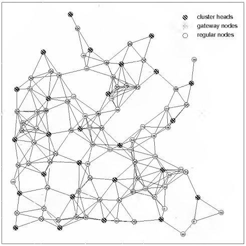

Figure 2: Clustering among 100 hosts, Transmission Power=100

up its clusterhead position according to lowest-id or highest-connectivity (or some other well defined priority scheme).

5 : Nodes which become separated from a cluster will recompute the clustering according to lowest-id or highest-connectivity.

Figure 1 shows that LCC further reduces clusterhead changes with respect to the existing schemes. We note that using LCC with lowest-id or highest-connectivity as the underlying mechanism does not make much difference.

Figure 2 shows a example of clustering using LCC with lowest-id among 100 nodes.

It should be pointed out that many other issues must be addressed in the design of a clustering algorithm with code separation across clusters. For example, as described in [6], use of a common control code for initialization and for reconfiguration; selection of orthogonal codes in adjacent clusters, etc. Specific solution to these problems are omitted here for brevity, but are reported in [7].

4.2 MAC layer

Clustering provides an effective way to allocate wireless channels among different clusters. Across clusters, we enhance spatial reuse by using different spreading codes (i.e. CDMA [8]). Within a cluster, we use a clusterhead controlled token protocol (i.e. polling) to allocate the channel among competing

nodes. The token approach allows us to give priority to clusterheads in order to maximize channel utilization and minimize delay. A clusterhead should get more chances to transmit because it is in charge of broadcasting within the cluster and of forwarding messages between mobile hosts which are not "connected". The channel access scheme is as follows:

1: Initially, the clusterhead gets the permission token to access the radio channel. It transmits any messages it has in its transmission queue.

2: The clusterhead passes the token to one of its neighbors according to a separately defined scheduling algorithm.

3: The cluster node (regular node or gateway) returns the token to its clusterhead after it has transmitted its message(s) (if any).

4: Repeat 1 to 3.

For each cluster only one node, which gets the permission token, can access the channel with an assigned code (CDMA). In some cases the permission token may be lost. One such case occurs when the node with the permission token moves outside the cluster. Another case is when the host is a gateway (which belongs to more than one cluster). The gateway might be tuned to a different code (i.e. different cluster), thus missing the permission token which is then lost. To overcome these problems, the clusterhead reissues the permission token after timeout.

We can use heuristic token scheduling algorithm (described in section 4.3) to choose the next neighbor host to get more efficient channel utilization and message delivery performance. Also we can reserve some channel accesses (more chances) for real time or multimedia traffics.

4.3 Gateways and pseudo links

We define a node as *gateway* if it belongs to more than one cluster. To communicate within a cluster, a gateway must select the code used by that cluster. We assume that a gateway can change its code after it returns the permission token or it receives a message. When a clusterhead issues the permission token to a gateway which is tuned to a different code, the token will be lost (i.e. a code conflict occurs). Clearly, code scheduling will affect the message delivery performance. In section 4.4, we will describe a heuristic code scheduling scheme to improve the message delivery. In addition, we will explore the performance improvement of multiple radio interfaces for gateways. Once a

gateway is equipped with multiple radio interfaces, it can access multiple cluster channels by selecting corresponding codes for each radio, thus reducing gateway conflicts.

Initially (as in figure 2) we define that there is a link between two nodes which can communicate with each other if these two nodes are within their transmission range and we have uniform power. After clustering, two nodes that share the same link may be unable to communicate since they use different codes for transmission. Links between two nodes, which use different codes, are defined as pseudo links and will be removed from connectivity. For example, in figure 2, link between node-90 and node-91 is a pseudo link.

In the cluster TDMA [6], clusters are created by using lowest-id scheme, and it needs globally synchronous slotted TDMA frame among all nodes. It is easy to reserve slots for real time traffic but it is difficult for practical implementation. The cluster token scheme [13] uses a more complex clustering (non-overlapped, non-clusterhead) and the token is circulated within a cluster in a predefined order. Both didn't take advantage of clusterheads. In our clusterhead token scheme, LCC clustering is more stable than lowest-id, and the token scheduling is more efficient.

5 Routing

Conventional routing protocols (distance vector or link-state) are not suitable for frequent, unpredictable change of topology. Link-state protocols are not appropriate for multihop mobile networks since they require that each node must know the entire network knowledge, which is not possible since topology is changing too rapidly for these algorithms to converge.

There are two extremes for mobile environment routing : *shortest-path* algorithm [2] which is suitable for a low rate of topology change, and *flooding* [2] which is suitable for a high rate of topology change. Flooding will increase communication overhead and shortest-path algorithms result in a need to maintain updated routing tables. Both increase the interference of channel access and degrade the throughput and response performance.

We want to use the facilities of clusterhead clusters and token scheduling to route packets in order to reduce the channel access overhead and improve message delivery.

To explore routing protocols which are suitable for cluster-token mobile wireless networks, we must understand the properties of transmitting a packet from one node to another. Token scheduling (in clusterheads) and code scheduling (in gateways) are the main factors affecting the routing efficiency. For

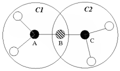

Figure 3: Routing example

example, in figure 3, to transmit a packet from node A to node C:

1: node A (clusterhead of $C1$) must get the permission to transmit (receives the token) in cluster $C1$.

2: node B (gateway) must select the same code as node A to receive the packet from node A.

3: node B must select the same code as node C (clusterhead of $C2$) and get the permission to transmit in cluster $C2$ (receives token from node C).

One important requirement in mobile networks is the avoidance of loops which are caused by stale routing tables. Several adaptive, loop free routing schemes have been recently proposed specifically for wireless, mobile networks [15,5]. In our proposed scheme we use as a basis the Destination Sequenced Distance Vector (DSDV) routing scheme [15] which was recently implemented also in cluster TDMA [6] and cluster token [13] schemes. DSDV stamps increasing sequence numbers on routing updates relative to a given destination. This way, stale updates can be easily detected and loops avoided.

Figure 4 shows the average delay versus various node mobility, in both free space channel and fading channel, for transmitting 100 packets from one node to another with an average distance of 12.31 hops in a 100-node network. We will compare the performance with the following protocols on the mobility of 1 m/s.

The average delay (simulation clock) of DSDV for the sample case is 36682 (in free space) . We can improve the routing efficiency by using some heuristic token scheduling and code scheduling schemes.

5.1 Cluster (hierarchical) Routing Protocol (DSCR)

In our project, we modify the DSDV scheme by exploiting the clusterheads. Namely, we use hierarchical routing to route packets. Each node maintains a cluster member table which records the destination clusterhead for each node, and broadcasts it periodically. A node will update its cluster member table

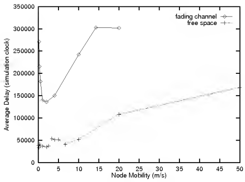

Figure 4: Average Delay of DSDV

when it receives a new one from its neighbor. Here again we use destination sequence numbers as in DSDV to avoid stale tables. There are two tables for each node to route packets. One is the cluster member table which is used to map destination address to the destination clusterhead address, and the other is the routing table which is used to select the next node to reach the destination cluster. When a node gets the permission token and transmission queue is not empty, it will first select the shortest (minimal hop) destination clusterhead according to the cluster member table and routing table, and then it will select the next node to transmit for that destination clusterhead according to the routing table. We call this cluster (hierarchical) routing scheme DSCR. The average delay of DSCR for the sample case is reduced to 35772 (in free space) (1.025 speedups over DSDV).

5.2 Clusterhead Gateway Switch Routing (CGSR) Protocol

One way to improve the routing efficiency is to route packets alternatively between clusterheads and gateways. That is, a packet will be routed via C_1, G_1,C_2,G_2,C_3,G_3.., where C_i are clusterheads and G_i are gateways, and finally reach its destination clusterhead. We call this routing scheme CGSR. Figure 5 shows routing examples for DSDV, DSCR, and CGSR.

Clusterhead gateway switch routing improves the routing efficiency since clusterheads have more changes to transmit and gateways are the only nodes that clusterheads can forward packets to.

When a node (clusterhead or gateway) becomes a regular node[a] , it will

[a] A regular node is a node which is neither a clusterhead nor a gateway

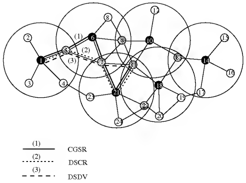

Figure 5: Routing examples (from node 1 to node 11)

forward all its packets to its clusterhead, thus returning to normal routing. The average delay of CGSR for the sample case is reduced to 32238 (in free space) (1.138 speedups over DSDV).

5.3 CGSR with priority token scheduling (CGSR+PTS)

In this clusterhead-token infrastructure, we can use various token schedule schemes to improve the routing efficiency. One way to do this is to give higher priority to neighbors from which a packet was recently received. The clusterhead gives the permission token to the upstream neighbor (gateway) in such a way that the packets will be sent with least delay. We call this routing scheme CGSR+PTS. Here is a simple way to implement priority-token-scheduling.

- Initially every neighbor of a clusterhead has the same priority to receive the token from the clusterhead.
- When a data packet is transmitted by node i, the clusterhead increases the priority of node i.
- When the token returns from an empty queue at neighbor j, the clusterhead decreases the priority of node j.

More generally, priority token scheduling allows us to forward high priority traffics with the least delay. Moreover, dynamic scheduling permits us to reserve a portion of the channel by offering more transmission opportunities to real time and multimedia sources.

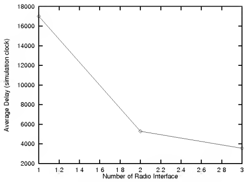

Figure 6: Average delay of CGSR+PTS+GCS with multiple radio in free space channel

Previous cluster oriented schemes, such as cluster TDMA [6] and cluster token [13], did not take full advantage of clusterheads. In our clusterhead oriented token scheme, the clusterhead plays an important role both in clustering and in dynamic channel scheduling. As a result, LCC clustering is more stable than previous clustering schemes, and token scheduling is more flexible.

It is easy for a clusterhead to forward (broadcast) packets to downstream nodes since a clusterhead has more chances to transmit and all its neighbors can receive the packets if their codes are selected correctly. The average delay of CGSR+PTS for the sample case is reduced to 22498 (in free space) (1.63 speedups over DSDV).

5.4 CGSR+PTS plus Gateway code scheduling (CGSR+PTS+GCS)

In the CGSR scheme, packets will be transmitted through clusterheads and gateways alternatively. Using CGSR+PTS which gives upstream nodes higher priority to get a permission token to forward packets to clusterhead nodes. On the other hand, we can use some heuristic code scheduling schemes for gateways to improve packets delivery from clusterheads to gateways. One better way to improve the forwarding is to use a more heuristic code scheduling than random scheduling. In this experiment, we give more priority to upstream clusterhead of a gateway after this gateway transmits a packet to its downstream clusterhead. The principle is that the gateway must switch its code to hear the upstream clusterhead in order to receive a packet after it sends out a packet to its downstream clusterhead. In the same way, the gateway will switch its code to the downstream clusterhead in order to receive the permis-

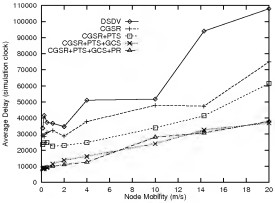

Figure 7: Average Delay of CGSR∗ and DSDV in free space channel

sion token to forward the packet after it receives a packet from its upstream clusterhead. We call this routing scheme CGSR+PTS+GCS. The average delay of CGSR+PTS+GCS for the sample case is reduced to 11414 (in free space)(3.214 speedups over DSDV).

Current radio interface technology can't switch code at will. Multiple radio interfaces provide another possible way to improve the gateways' performance. With multiple radio interfaces, a gateway can access multiple clusters simultaneously by using different codes. Figure 6 shows the simulation result of multiple radio interfaces. For best cost/performance tradeoff, two radio interfaces is the best choice.

5.5 CGSR+PTS+GCS plus path reserving (CGSR+PTS+GCS+PR)

In CGSR+PTS and CGSR+PTS+GCS, we give more priority to the upstream node (clusterhead and gateway) to improve the packets' forwarding efficiency. These schemes will perform well if the upstream node does not change much. The upstream node will change if it finds another shorter path to the destination clusterhead. To keep the path more stable for CGSR+PTS and CGSR+PTS+GCS, we can reserve the path until it is disconnected. Once the first packet selects the next node to route, it will keep this path until it breaks (becomes disconnected or pseudo link). Reserving the next hop for successive packets' transmission facilitates the CGSR+PTS+GCS routing scheme and makes multimedia traffics (audio and video) viable in multihop, mobile networks. We call this routing scheme CGSR+PTS+GCS+PR. The average

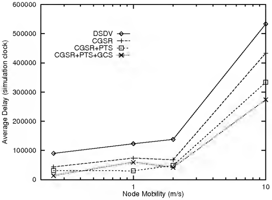

Figure 8: Average Delay of CGSR$*$ and DSDV in fading channel

delay of CGSR+PTS+GCS+PR for the sample case is reduced to 9877 (in free space)(3.714 speedups over DSDV).

Figure. 7 shows the average delay improvement of CGSR$*$ [b] schemes over DSDV for different mobility. CGSR$*$ greatly improve the packet delivery in the clusterhead-token infrastructure and CGSR+PTS+GCS$*$ perform well for higher mobility. When the mobility is slow, CGSR+PTS+GCS+PR provides a better reservation, thus reducing the average delay.

Figure 8 shows the case of fading channel model. In fading channel model, the performance will degrade due to the noise channel and retransmission. The simulation time for fading channel model is greatly larger than the free space model. It takes 20 hours to run a general case. To speedup the simulation time and reduce the turnaround time, we port the simulator from a sequential to a parallel environment [14]. Experimental result shows that speedup up to 10 is reached under 16 processors. Figure 9 shows the speedup of parallel simulation for various configuraitons.

6 Conclusion

A clusterhead-token infrastructure has been proposed for multihop, mobile wireless networks. Least cluster change (LCC) clustering provides the stablest cluster structure for grouping mobile nodes and allocating radio channel codes. Clusterhead controlled token protocol allocates channel access within

[b] CGSR$*$ means CGSR, CGSR+PTS, CGSR+PTS+GCS, and CGSR+PTS+GCS+PR.

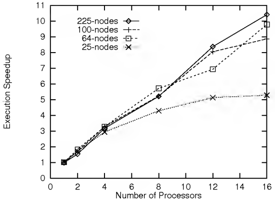

Figure 9: Speedup of parallel simulation in fading channel

a cluster and facilitates packets forwarding. The clusterhead gateway switch routing (CGSR) schemes deliver packets efficiently and make clusterhead token scheduling and gateway code scheduling valuable. Heuristic token scheduling and gateway code scheduling speed packets delivery along multihop paths. Path reservation makes token scheduling and code scheduling schemes more efficient, thus being capable of transmitting multimedia traffics. Simulations in free space and fading channel model have been built, and parallel simulation provides an efficient way for large and complex wireless networks.

Acknowledgments

This research was supported by the U.S. Department of Justice/Federal Bureau of Investigation, ARPA/ITO under Contract J-FBI-93-112 Computer Aided Design of High Performance Network Wireless Networked Systems, and by Intel under project "QoS Wireless Networks."

References

1. R. Bagrodia and W. Liao, *Maisie User Manual*, Computer Science Department, University of California at Los Angeles, (1993).
2. D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc., pp.297-333, (1987).
3. M. Scott Corson, A. Ephremides : *A Distributed Routing Algorithm for Mobile Wireless Networks*, ACM Journal on Wireless Networks, Vol.1,

No.1, (1995).

4. A. Ephremides, J.E. Wieselthier and D.J. Baker, *A design concept for reliable mobile radio networks with frequency hopping signaling*, Proc. IEEE 75, pp.56-73, (1987).
5. J. J. Garcia-Luna-Aceves, *A unified approach to loop-free routing using distance vectors or link states*, ACM SIGCOM'89, pp.212-223, (1989).
6. Mario Gerla and Jack Tzu-Chieh Tsai : *Multicluster, mobile, multimedia radio network*, ACM-Baltzer Journal of Wireless Networks, Vol.1, No.3, pp.255-265, (1995).
7. Mario Gerla and Ching-Chuan Chiang, *Multicast routing in multihop, mobile wireless networks*, UCLA-CSD technical report, Sept. (1996).
8. K.S. Gilhousen, I. M. Jacobs et al., *On the capacity of a cellular CDMA system*, IEEE Trans. Veh. Tech. Vol.40, pp.303-312, (1991).
9. M. Gudmundson, *Correlation Model For Shadow Fading In Mobil Radio Systems*, Electron. Lett., pp.2145-2146, (1991).
10. John Ioannidis, Dan Duchamp, Gerald Q. Maguire Jr : *IP-based protocols for mobile internetworking*, Columbia University, ACM SIGCOMM Symposium on Communication, Architecture and Protocols, Zürich, Suisse, September, pp.235-243, (1991).
11. David B. Johnson, *Routing in Ad Hoc networks of mobile hosts*, Proc. of Workshop on Mobile Computing and Applications, Dec. (1994).
12. J. Jubin and J. D. Tornow, *The DARPA packet radio network protocols*, Proc. of the IEEE, Vol.75, No.1, Feb., (1987).
13. Chunhung Richard Lin and Mario Gerla,*A distributed architecture for multimedia in dynamic wireless networks*, IEEE International Conference on Communications (ICC'95), pp.1468-1472, (1995).
14. Winston W. Liu, Ching-Chuan Chiang, et al., *Parallel simulation environment for mobile wireless networks*, Proc. of the 1996 winter simulation conference, Dec., (1996).
15. C.E. Perkins, and P. Bhagwat, *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*, ACM SIGCOMM'94, pp.234-244, (1994).
16. T. S Rappaport, and S. Y. Seidel, SIRCIM: Simulation of Indoor Radio Channel Impulse Response Models, *VTIP, Inc.*, (1990).